What is claimed is:

1. A computer-implemented method for increasing the performance of a virtual machine, the computer-implemented method comprising:

obtaining a program instruction to be executed by the virtual machine;

determining when the program instruction is a branch instruction;

determining when a basic block is present in a code cache when it is determined that the program instruction is a branch instruction, the basic block including code, the code cache being associated with the virtual machine; and

executing the code included in the basic block when it is determined that the basic block is present.

2. A computer-implemented method as recited in claim 1 wherein when it is determined that the basic block is not present in the code cache, the method further includes:

interpreting the program instruction; and

copying code corresponding to the program instruction into the code cache.

3. A computer-implemented method as recited in claim 2 further including:

allocating space in the code cache for the code corresponding to the program instruction; and

providing the code corresponding to the program instruction with a label.

4. A computer-implemented method as recited in claim 3 further including placing the label in a table of labels.

5. A computer-implemented method as recited in claim 2 wherein determining when the basic block is present in the code cache includes searching through a table of labels to determine if a target associated with the program instruction has a substantially matching label in the table of labels.

6. A computer-implemented method as recited in claim 2 wherein the program instruction is a bytecode, and wherein the bytecode is executed by an interpreter of the virtual machine.

7. A computer-implemented method as recited in claim 2 wherein the code cache is a native code cache, and the code corresponding to the program instruction is native code.

8. A computer-implemented method as recited in claim 1 wherein the program instruction is a bytecode and the code cache is a native code cache.

9. A computer-implemented method as recited in claim 8 further including interpreting the bytecode when it is determined that the program instruction is not the branch instruction.

10. A computer-implemented method as recited in claim 1 further including:

computing a target using the program instruction, wherein determining when the basic block is present in the code cache includes determining if the code cache includes any basic blocks which correspond to the target.

11. A computer program product for increasing the performance of a virtual machine, the computer program product comprising:

computer code for obtaining a program instruction to be executed by the virtual machine;

computer code for determining when the program instruction is a branch instruction;

computer code for determining when a basic block is present in a code cache when it is determined that the program instruction is a branch instruction, the basic block including code, the code cache being associated with the virtual machine;

computer code for executing the code included in the basic block when it is determined that the basic block is present; and

a computer-readable medium that stores the computer codes.

12. A computer program product as recited in claim 11 further including:

computer code for interpreting the program instruction when it is determined that the basic block is not present in the code cache; and

computer code for copying code corresponding to the program instruction into the code cache when it is determined that the basic block is not present in the code cache.

13. A computer program product as recited in claim 12 further including:

computer code for allocating space in the code cache for the code corresponding to the program instruction;

computer code for providing the code corresponding to the program instruction with a label; and

computer code for placing the label in a table of labels.

14. A computer program product as recited in claim 11 wherein the program instruction is a bytecode and the code cache is a native code cache.

15. A computer program product as recited in claim 11 wherein the computer-readable medium is one selected from the group consisting of a data signal embodied in a carrier wave, a floppy disk, a computer memory, a hard disk, an optical disk, a tape drive, and a CD-ROM.

16. A computing system, the computing system including a virtual machine, the virtual machine comprising:

a code cache; and

an interpreter, the interpreter being arranged to obtaining a bytecode, the interpreter further being arranged to determining when the bytecode is a branch bytecode and to determine when a basic block is present in the code cache when it is determined that the bytecode is a branch bytecode, the basic block including native code, wherein the interpreter causes the native code to be executed when it is determined that the basic block is present.

17. A computing system including a virtual machine according to claim 16 wherein the interpreter is further arranged to interpret the bytecode when it is determined that the basic block is not present in the code cache and to copy native code corresponding to the bytecode into the code cache.

18. A computing system including a virtual machine according to claim 16 wherein the interpreter is further arranged to interpret the bytecode when it is determined that the bytecode is not a branch bytecode.

19. A computer-implemented method for increasing the performance of an interpreter, the method comprising:

identifying a portion of compiled code which corresponds to a block of source code; and

copying the portion of compiled code into a code cache.

20. A computer-implemented method as recited in claim 19 wherein the code cache is a native code cache, and the portion of compiled code is native code.

21. A computer-implemented method as recited in claim 20 further including:

executing the portion of compiled code copied into the native code cache using a processor.

22. A computing system comprising:

a processor;

an interpreter; and

a native code cache, the native code cache being associated with the processor, wherein the interpreter includes at least one block of binary code that is arranged to be copied substantially directly into the native code cache.

23. A computing system according to claim 22 wherein the at least one block of binary code is associated with a bytecode which is arranged to be interpreted by the interpreter.

24. A computing system according to claim 23 wherein the at least one block is arranged to include indirect calls.